

PCTWORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : H04B	A2	(11) International Publication Number: WO 97/08838 (43) International Publication Date: 6 March 1997 (06.03.97)
(21) International Application Number: PCT/US96/12958 (22) International Filing Date: 12 August 1996 (12.08.96) (30) Priority Data: 08/514,736 14 August 1995 (14.08.95) US (71) Applicant (for all designated States except US): ERICSSON INC. [US/US]; 7001 Development Drive, P.O. Box 13969, Research Triangle Park, NC 27709 (US). (72) Inventors; and (75) Inventors/Applicants (for US only): AUTRY, George, M., IV [-/US]; 8 Dogwood Place, Lynchburg, VA 24502-3814 (US). HOGE, Philip, M. [-/US]; 199 Peters Drive, Forest, VA 24551 (US). LUTTRELL, Edward, N. [-/US]; 412-B Kerry Lane, Lynchburg, VA 24502 (US). RODERIQUE, William, J. [-/US]; 4413 Goodview Street, Lynchburg, VA 24502 (US). RONDEAU, Michael, N. [-/US]; 101 Meadowgate Drive, Forest, VA 24551 (US). THOMSON, Bradley, A. [-/US]; 144 St. Johns Drive, Lynchburg, VA 24503 (US). (74) Agent: LASTOVA, John, R.; 8th floor, 1100 North Glebe Road, Arlington, VA 22201-4714 (US).		(81) Designated States: AL, AM, AT, AU, AZ, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, US, UZ, VN, ARIPO patent (KE, LS, MW, SD, SZ, UG), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG). Published <i>Without international search report and to be republished upon receipt of that report.</i>
(54) Title: METHOD AND APPARATUS FOR MODIFYING A STANDARD INTERNETWORK PROTOCOL LAYER HEADER		
(57) Abstract <p>In a packet-base communications system where each packet includes a header and an associated data portion, message packets are communicated between devices on first and second networks through a gateway. The first network uses a standard internetwork protocol (IP). Predetermined fields in the standard (IP) header of each message packet are eliminated to obtain a modified header before transmitting message packets over the second network to the second device. Conversely, one or more predetermined fields are added to the modified header of each packet in another message from the second device to the first device to convert that modified header into a corresponding standard IP header before transmission over the first network. In a specific embodiment, the present invention permits internetwork communications between computers in a radio frequency communications network and computers connected to a wireline network compatible with conventional TCP/IP protocols. Unnecessary IP header bits are removed before packets are transmitted over the radio network to conserve RF channel bandwidth. Knowledge of information already present on the data link layer of the RF channel communications protocol is used to omit unnecessary or redundant fields in the standard IP header of the network layer. Enough information is preserved in the reduced IP network header so that the standard IP header may be reassembled/reconstructed.</p>		

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia	GB	United Kingdom	MW	Malawi
AT	Austria	GE	Georgia	MX	Mexico
AU	Australia	GN	Guinea	NE	Niger
BB	Barbados	GR	Greece	NL	Netherlands
BE	Belgium	HU	Hungary	NO	Norway
BF	Burkina Faso	IE	Ireland	NZ	New Zealand
BG	Bulgaria	IT	Italy	PL	Poland
BJ	Benin	JP	Japan	PT	Portugal
BR	Brazil	KE	Kenya	RO	Romania
BY	Belarus	KG	Kyrgyzstan	RU	Russian Federation
CA	Canada	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic	KR	Republic of Korea	SE	Sweden
CG	Congo	KZ	Kazakhstan	SG	Singapore
CH	Switzerland	LI	Liechtenstein	SI	Slovenia
CI	Côte d'Ivoire	LK	Sri Lanka	SK	Slovakia
CM	Cameroon	LR	Liberia	SN	Senegal
CN	China	LT	Lithuania	SZ	Swaziland
CS	Czechoslovakia	LU	Luxembourg	TD	Chad
CZ	Czech Republic	LV	Larvia	TG	Togo
DE	Germany	MC	Monaco	TJ	Tajikistan
DK	Denmark	MD	Republic of Moldova	TT	Trinidad and Tobago
EE	Estonia	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	UG	Uganda
FI	Finland	MN	Mongolia	US	United States of America
FR	France	MR	Mauritania	UZ	Uzbekistan
GA	Gabon			VN	Viet Nam

**METHOD AND APPARATUS FOR MODIFYING A
STANDARD INTERNETWORK PROTOCOL LAYER HEADER**

FIELD OF THE INVENTION

5

The present invention relates to transmission of data packets over an internetwork, and more particularly, to reducing the bandwidth occupied by headers associated with those data packets over a portion of the internetwork.

10

BACKGROUND AND SUMMARY OF THE INVENTION

In an attempt to establish a standard communications model, the international organization for standardization generated in the late 70's the open systems interconnection (OSI) model for computer-to-computer dialogue. The OSI model divides the communications process into seven layers illustrated for example in Figure 1(a). Certain communication tasks are assigned to certain layers and the output of each layer has a precise format established for it. As illustrated in Figure 1(a), data from an application or process running on a first host computer (host A) passes through each OSI layer on its way to the communications network. As the information descends through each layer, it undergoes a transformation that prepares it for processing by the next layer. Upon reaching the bottom layer, data is passed to the network as a serial stream of bits represented by a changing signal, i.e. changing voltages, microwaves

or light pulses. After receiving ^{the changing signal at} host B, the stream travels in reverse order through the seven OSI layers.

In general, the application layer is the only part of the OSI
5 model the computer user sees as it ^{full} fills requests ~~to work with~~ ^{using} various application programs and ~~with~~ information, such as a shared database, that may be found either in the local host or in a host elsewhere on the network. The presentation layer ensures that the message takes a form that the receiving host can interpret and may
10 include language translation, data compression and/or data encryption. The session layer opens communication with the receiving host and determines whether the two hosts A and B will be able to address each other simultaneously (full duplex) or must take turns (half duplex). Here the session layer brackets the message by
15 marking its beginning and end. ^{using} The transport layer breaks the message into segments keeping track of their sequence in the message. Each segment is assigned a checksum. After dividing the message into segments, the transport layer makes a backup copy of each segment should the original be damaged or destroyed in route.
20 After a message segment enters the network layer, it is divided into packets, and a route is selected for the message. In the data link layer, a checksum is calculated for each message packet, and a link layer address of the next destination of the packet in its route is added to the packet. In the physical layer, the bits of each packet
25 are encoded onto an analog signal sent over the communications medium, e.g., telephone lines.

Layer protocols and interfaces therebetween are defined which provide specifications for communication between a process or program being executed on one host computer's operating system and another process or program running on another computer.

- 5 Transmission control protocol/internet protocol (TCP/IP) are two protocols that are part of a protocol suite or family of protocols layered and designed to connect computer systems that use different operating systems and network technologies. TCP/IP, which provides a common set of protocols for invocation on dissimilar interconnected
10 systems, is shown in Figure 1(b) mapped to analogous layers of the OSI model.

- TCP/IP is a four layer protocol suite which facilitates the interconnection of two or more computer systems on the same or
15 different networks and in certain networks, such as the Internet, is a requirement for interoperability. The four layers comprise two independent protocols: TCP, which can be used to access applications on other systems within a single network, and IP, which permits identification of source and destination addresses for communication
20 between systems on different networks. The present invention is directed to the latter IP protocol.

- At the internet layer, the internet protocol permits identification of source and destination addresses for communication
25 over physical networks. The fundamental internetwork service consists of a packet delivery system, and the internetwork protocol (IP) defines that delivery mechanism, i.e., the basic unit of data

transfer. Thus, the IP specifies the exact format of all data as it passes across a TCP/IP internet. The IP software also performs a routing function by choosing a path over which data will be sent.

5 The basic data transfer unit is often called a "datagram" (and similar to a typical physical network "frame") and is divided into header and data areas. The datagram is shown in Figure 2(a). The header contains (1) source and destination addresses and (2) a type field that identifies the contents of the datagram. The IP only
10 specifies the header format including the source and destination IP addresses; it does not specify the format of the data area.

Figure 2(b) shows a standard IP header consisting of a number of predefined fields. The first four bit field VERS contains the
15 version of the IP protocol that was used to create the datagram. The VERS field is used to verify that the sender, receiver, and any gateways in between, agree on the format of the datagram. The four bit header length field HLEN provides the datagram header length measured in multiples of thirty-two bit words. The TOTAL LENGTH
20 field gives the length of the IP datagram (both header and data) measured in octets. By subtracting the length of the header from the total length found in the TOTAL LENGTH field, the size of the data area can be calculated. The eight bit SERVICE type field specifies how the datagram should be handled.

25

The three fields--IDENTIFICATION, FLAGS, and FRAGMENT OFFSET -- control fragmentation and reassembly of datagrams.

Since an IP datagram may not fit into one physical frame data area, (a constraint of the physical network), the datagram may be divided into smaller pieces called "fragments" with the process of dividing a datagram being known as "fragmentation." Fragment size is chosen
5 so that each fragment can be shipped across the underlying network in a single frame. Each fragment has the same format as the original datagram. Accordingly, each fragment contains a header that duplicates most of the original datagram header except for a bit in the FLAGS field. The FLAGS bit identifies the datagram as a
10 fragment which carries a data amount up to a total length that is smaller than the maximum transfer unit permitted over the network. Once a datagram is fragmented, the fragments travel as separate datagrams all the way to the ultimate destination where they must be reassembled. The field IDENTIFICATION contains a unique
15 integer that identifies the datagram to allow the destination to know which of the arriving fragments belong to which datagrams. As the fragment arrives, the destination uses the IDENTIFICATION field along with the datagram source address to identify the datagram. The FRAGMENT OFFSET specifies the offset in the original
20 datagram of the data being carried in the fragment measured in units of eight octets starting at offset zero. To reassemble the datagram, the destination must obtain all fragments starting with the fragment that has offset "zero" through the fragment with the highest offset. The FLAGS field controls fragmentation with one of
25 the bits being called the "more fragments bit."

The TIME TO LIVE (TTL) field specifies how long in seconds the datagram is allowed to remain in the Internet system and is normally used as a count value so that time does not have to be synchronized across the network. The PROTOCOL field specifies
5 which high level protocol was used to create the message being carried in the data area of the datagram. In essence, the value of the PROTOCOL field specifies the format of the data area. The field HEADER CHECKSUM is formed by treating the header as a
10 sequence of sixteen bit integers, adding them together using one's complement arithmetic, and taking the one's complement of the result. Fields SOURCE IP ADDRESS and DESTINATION IP ADDRESS contain the thirty-two bit IP addresses of the datagram sender and intended recipient. Thus, the IP addresses specify the original source and ultimate destination. The IP OPTIONS and
15 PADDING fields are included mainly for testing and debugging of the network.

Since each layer of the OSI model adds header type information, there is a considerable amount of overhead information
20 added to the originally transmitted data. This is in addition to the significant processing overhead associated with packetizing data for network and internetwork transmission. Because the network and internetwork packet header is meant to be used across a variety of applications, the standard IP packet header contains fields that may
25 not be used or needed in a particular application. While these extra fields may not be a problem in high bandwidth communication environments, they present a significant problem when used in a

radio frequency (RF) communications environment. RF bandwidth is typically a precious resource, e.g., as land mobile radio and cellular radio communications. Any additional overhead bits/fields decreases the amount of actual data that can be sent in the given unit of time
5 therefore lowering data throughput. Thus, an important objective of the present invention is to minimize the overhead required to send a packet of data over an RF data communications network but at the same time permit those RF data communications to be compatible with industry accepted internetwork protocol standards like IP and
10 TCP/IP.

In general, the present invention permits internetworked communications between computers in a radio frequency communications network and computers connected to a wireline,
15 minimizing ^{ed} protocol overhead ^{while} maintaining compatibility with conventional TCP/IP protocols. Unnecessary IP header bits are removed before packets are transmitted over the radio network to conserve RF channel bandwidth. Knowledge of information already present in the data link layer of the RF channel communications
20 protocol is used to omit unnecessary or redundant fields in the standard IP header of the network layer. Enough information is preserved in the reduced IP network header so that the standard IP header may be reassembled/reconstructed.

network ref to with

25 The present invention provides a method for generating a modified network layer header adapted from a standard network layer header by omitting certain fields included in the standard

network layer header. In a packet-based communications system, packeted messages are communicated between devices on first and second networks. The first network uses a standard internetwork protocol (IP). When a message is transmitted from a first device on
5 the first network to a second device on the second network, predetermined fields from the standard internetwork protocol field of each packet in the message are omitted to obtain a minimized header before transmitting the message over the second network to the second device. Predetermined fields are added to minimized headers
10 of each packet in messages from the second device to the first device to convert that minimized header into a corresponding standard IP header before transmitting the other message over the first network. A corresponding standard IP header is reconstructed by adding
known static, standard IP header fields to fields present in the
15 minimized header of each packet along with dynamic fields of the standard IP header determined from information contained in that minimized header.

The present invention provides a communications system that
20 permits first and second data processing units associated respectively with a first network and a second network where the second network has a more limited communications bandwidth than the first network to transmit packets of information. In one embodiment, the first
network may be a wireline network such as an Ethernet network,
25 and the second network may be a radio communications network which uses wireless radio frequency (RF) communications channels. A gateway connects the Ethernet and radio networks. For messages

received from the Ethernet intended for the radio network, the gateway omits certain fields of a standard IP header for each message packet to obtain a modified header. For messages received from the radio network intended for the Ethernet, the gateway adds
5 fields to the modified header to obtain a standard IP header.

BRIEF DESCRIPTION OF THE DRAWINGS

These objects and advantages, as well as others will be better
10 understood from the following detailed description taken in conjunction with the accompanying drawings in which:

FIGURE 1(a) is a diagrammatic view of an open systems
interconnection (OSI) model;
15

FIGURE 1(b) is a diagrammatic view of the OSI model of
Figure 1(a) compared with a transmission control protocol/internet
protocol (TCP/IP) model;

20 FIGURE 2(a) is a diagrammatic view of a standard internet
protocol (IP) datagram;

FIGURE 2(b) is a diagrammatic view of a standard internet
protocol (IP) header;
25

FIGURE 3 is a function block diagram of two unconnected
networks including an Ethernet network and an RF data network;

FIGURE 4 is a function block diagram of an internetwork that includes an Ethernet network coupled to an RF data network through a gateway;

5 FIGURE 5 is a function block diagram showing in more detail the gateway illustrated in Figure 4;

FIGURE 6 is a diagram showing a typical protocol stack with a standard network layer modified by a network driver in accordance
10 with the present invention;

FIGURE 7 is a flowchart diagram illustrating a procedure for generating a radio network header packet header from an IP packet header at the data gateway;
15

FIGURE 8 is a flowchart diagram illustrating the procedures for converting a radio network layer packet header to an IP packet header at the radio data terminal;

20 FIGURE 9 is a flowchart diagram illustrating the procedures for converting an IP packet header to a radio network layer packet header at the radio data terminal; and

FIGURE 10 is a flow chart diagram illustrating the procedures
25 for converting a radio network layer packet header to an IP packet header at the data gateway.

DETAILED DESCRIPTION OF THE INVENTION

In the following description, for purposes of explanation and not limitation, specific details are set forth such as particular circuits, protocols, techniques, etc. in order to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that the present invention may be practiced in other embodiments that depart from these specific details. In other instances, detailed descriptions of well-known methods, devices, protocols, and circuits are omitted so as not to obscure the description of the present invention with unnecessary detail.

Figure 3 shows two different types of networks including an Ethernet network 12 and an RF data network 10. No single type of network is best for all situations/applications. Ethernet networks perform well when used to connect computers physically located at the same location. However, an RF data network is best suited for mobile/portable radio data terminals (RDTs) where communications with other radio data terminals (and as described below, other host computers) are over a wireless, radio frequency (RF) physical communications medium. Host computers A, B, and C (16, 18, and 20) use Ethernet cable 14 as their physical communications mediums and use Ethernet addresses and protocol to communicate. The radio data terminals 30, 32, and 34 use radio frequencies as their physical communications medium and RF data network addresses and protocols to communicate between a central site 22 and radio/radio data interfaces 24, 26, and 28 corresponding to each radio data terminal 30, 32, and 34. The present invention relates to a gateway

that permits a radio data terminal on the RF data network to communicate with a host computer on a wireline network such as Ethernet network 12.

5 Figure 4 illustrates an Ethernet network 12 and the radio data network 10 internetworked using a data gateway 40 and a data interface module 38. In general terms, data gateway 40 provides the interface for translating messages between both networks. To
10 simplify internetworking in the OSI model, a network layer is used above the data link layer to provide consistent addressing, protocol, and interface across the internet. Although the network layer
 address provides a consistent address across the internet, it must be converted to a data link layer address specific to the network type to
 actually send data across a specific network. The data gateway 40
15 connects to the host computers 16, 18, and 21 on Ethernet network coaxial cable 14 using non-proprietary, standardized protocols such as TCP/IP. To interface with the RF data network, the data gateway 40
 supports network driver software that functions at both the network and data link layers to make necessary protocol conversions as
20 described in more detail below.

 Figure 5 illustrates the system architecture of the data gateway 40 and its external interfaces. One or more radio system
 interface (RSI) boards 64 and 66 handle communications to the radio
25 system. The central activity processor (CAP) 50 provides the IP Ethernet interface to host computers 18-20 over Ethernet coaxial
 cable 14. The central activity processor 50 also provides system services such as input and output of information to fixed disk 56 and

floppy disk 58 over a small computer system interface (SCSI) bus 54 and output of information to be printed using optional printer 60. The central activity processor 50 and radio system interfaces 64 and 66 communicate together over a conventional VME system bus 52.

5 The RSIs 64 and 66 exchange control messages over control link 74 to RF site controllers at radio sites 22 and 23 at, for example, 19.2 kbps or 9.6 kbps. The trunked system interfaces 64 and 66 send data to the radio site and receive data from the radio site using Rockwell modems 68 and 70 operating at, for example, 9600 bits per second

10 (bps). Each radio system interface provides four communication ports with each communication port handling one data call at a time.

Referring to Figure 4, the following describes the data flow from host computer 16 on the wireline Ethernet Network to radio

15 data terminal 30 on the RF Data Network.

1. The host 16 sends a message to the data gateway 40.
2. The data gateway 40 sends a call request to the data interface module 38.
3. The data interface module 38 sends the call request to the
- 20 site 22 where the radio 24 is located.
4. The site 22 instructs the radio 24 over an RF control channel to which the radio is tuned to tune to an RF working channel to receive the message.
5. The site 22 returns the RF working channel assignment to
- 25 the data interface module 38. The data interface module 38 connects a data path between the data gateway 40 and the working channel and notifies the data gateway 40.

6. The data gateway 40 breaks the message down into packets and sends the first burst of packets to the site 22. The site 22 forwards the burst to the radio as it receives it. After the radio 24 receives the entire burst, it sends an ACK back to the data gateway 40 (via the site 22), informing it of the packets that the radio correctly received. If necessary, the data gateway 40 sends another burst containing packets that the radio did not correctly receive and packets that the data gateway 40 has not previously sent. This sequence continues until the radio receives the entire message or until the data gateway 40 exhausts a preset number of retries.

7. The radio 24 sends the message to its RDI. If the message is large enough, the radio sends the initial part of the message to the RDI while the radio is still receiving the message from the data gateway 40.

8. The RDI acknowledges to the radio that it has successfully received the message.

9. The RDI forwards the message to the RDT.

The following describes the data flow from RDT 30 on the RF Data Network to host computer 16 on the wireline, Ethernet Network.

1. The Radio Data Terminal (RDT) 30 begins transferring a message to its Radio Data Interface (RDI).

2. The RDI begins pipelining the message to the radio 24 using a radio signaling protocol.

3. The RDI acknowledges to the RDT 30 successful receipt of the message.

4. The radio 24 informs the site 22 over the RF control channel that it has a message and requests an RF working channel.

5. The site 22 assigns an available RF working channel and informs the radio 24.

5 6. The site 22 sends the call assignment to the data interface module 38 which sends it on to the data gateway 40.

7. The data gateway 40 selects a radio system interface Port and informs the data interface module 38. The data interface module 38 sets up a data path between the data gateway 40 and the RF
10 working channel.

8. The radio 24 acknowledges to the RDI that it has successfully received the message.

9. The radio 24 breaks the message down into packets and sends the first burst of packets to the site 22. The site 22 forwards
15 the burst to the data gateway 40 as it receives it. After the data gateway 40 receives the entire burst, it sends an ACK back to the radio, informing it of the packets that the data gateway 40 correctly received. If necessary, the radio sends another burst containing
20 packets that the data gateway 40 did not correctly receive and packets that the radio has not previously sent. This sequence continues until the data gateway 40 receives the entire message or until the radio exhausts its retries.

10. The radio 24 tells the RDI the status of the message transmission to the data gateway 40.

25 11. If the data gateway 40 successfully received the message, the data gateway 40 sends the message to the host computer 16. The message transfer from the data gateway 40 to the host computer 16 proceeds independently of any other signaling from the RDT.

12. If requested, the RDI tells the RDT whether the EDG successfully received message or not.

In communicating with the Ethernet network, the data link
5 layer of the data gateway uses Ethernet II protocol also known as IEEE 802.3 DIX. The network layer uses the standard internet protocol (IP) described, for example, in the text by Douglas E. Comer entitled "Internetworking with TCP/IP, Vol. 1." Used above the network layer are "host-to-host" conversations between a host
10 computer 16-21 and a radio data terminal 30-36 which follow transport layer protocols. Any headers that they use are simply passed as data through the network and are not of particular interest to the present invention.

15 At the data link layer, the data gateway 40 and host computers 16-21 communicate using Ethernet addresses and address resolution protocol (ARP) to discover each other's Ethernet addresses based on their IP addresses. The network layer uses the IP address to decide where to route the message next. For messages originating
20 from a host computer, the host computer addresses a radio (or group of radios) using a unique IP address assigned to each radio (and group). Normally, each host computer has a single entry added to its routing table instructing it to use the data gateway central activity processor 50 as a next gateway for messages being sent to any
25 destination on the RF data network 10. For messages from a radio to a host, the data gateway 40 receives the message, examines the IP address, and forwards the message on to the appropriate host computer.

The data link layer protocol between the data gateway 40 and the radio/radio data interfaces 24-28 is a non-standard, hardened protocol designed specifically for limited bandwidth RF working channels assigned by sites 22 or 23. The radio data terminals 30-36
 5 physically connect to the radio/radio data interfaces 24-29 using, for example, a 9600 bps synchronous serial link. The data link layer on the radio network uses a medium access control sublayer network driver designed for personal computers running MS-DOS which converts between standard IP headers and RF data network layer
 10 headers. Figure 6 illustrates this function of the network driver at the radio data terminal to perform data link layer functions between the radio data terminal and the data gateway.

Standard Internet Protocol (IP) Headers are converted to Radio
 15 Network Layer (RNL) Headers before messages are sent across the Radio Data Network. Modified (smaller) Radio Network headers are created by omitting the highlighted information in the standard IP header and using the unhighlighted fields.

20	Version	Header Length	Service Type
	Total Length		
	Identification		
	D F	U F	M F Fragment Offset
25	Time To Live		Protocol

18

5	Header Checksum	
	1st 16 bits of Source IP address	
	Last 16 bits of Source IP Address	
	1st 16 bits of Destination IP Address	
	Last 16 bits of Destination IP Address	
	1st 16 bits of IP Options	
	Last 8 bits of IP Options	Padding

- 10 The Radio Network Layer Header shown below is stripped of unnecessary fields for the radio interface between the RDTs and the data gateway 40.

15	Vers		Unused	
	Identification			
	U	U	M F	Fragment Offset
	1st 16 bits of Extended Address			
	Last 16 bits of Extended Address			
20	Protocol			

The following is a description of the RNL header fields:

VERS = The version of the radio network layer header used to create the datagram.

IDENTIFICATION = Number that uniquely defines all of the
5 fragments of the same message from the same source.

U = Unused Bit. Available for future use.

MF = More Fragments. This bit is set if there are more fragments
10 coming in the current message.

FRAGMENT OFFSET = This field tells where this fragment belongs in the current message. When a message is fragmented, each fragment except the last one must be a multiple of 8 bytes long. The
15 fragment offset is multiplied by 8 to determine the byte offset. A maximum message size may be for example 64K bytes.

EXTENDED ADDRESS = For messages to the radio network, this field defines the Source IP Address; for messages from the radio
20 network, this field defines the Destination IP Address.

PROTOCOL = This field is passed through as defined by the Standard IP Header.

25 If the IP datagram has 512 bytes or less of data, the More Fragments (MF) and FRAGMENT OFFSET fields are copied from the IP Header. If the IP datagram has more than 512 bytes of data and the datagram is destined for a radio, the data gateway 40 fragments

the datagram and uses the appropriate values in the MF and
 FRAGMENT fields. For datagrams from the data gateway 40, the
 Extended Address is the SOURCE IP Address. For datagrams from
 a Radio Data Terminal (RDT), the Extended Address is the
 5 DESTINATION IP Address.

Radio Network Layer Headers are converted in the data
 gateway 40 to IP headers after datagrams are sent across the Radio
 Network. The RNL Header is transformed into an IP Header from
 10 the following information:

Static Fields used in all IP Headers:

	VERSION	0	U bit (unused)	0
	HEADER LENGTH	5	TIME TO LIVE	255
15	SERVICE TYPE	0	IP OPTIONS	Not Used
	DF bit	0	PADDING	Not Used

Configured Information:

The IP Address of the Radio.

20

Fields Gathered from the Radio Network Data Link Layer:

TOTAL LENGTH is calculated from the data link layer total
 length plus 10 bytes. The 10 bytes correspond to extra data added
 25 for the conversion from the radio network layer header to the IP
 header.

5	1st 16 bits of Source IP address	
	Last 16 bits of Source IP Address	
	1st 16 bits of Destination IP Address	
	Last 16 bits of Destination IP Address	
	1st 16 bits of IP Options	
	Last 8 bits of IP Options	Padding

When the data gateway 40 receives a datagram from an RDT,
 10 the data gateway converts the Data Link Layer Address to the
 Source IP Address using configuration information in the data
 gateway. The data gateway uses the Extended Address in the RNL
 Header as the Destination IP Address. When an RDT receives a
 datagram from the data gateway, the RDT uses the Extended
 15 Address in the Radio Network Layer Header as the Source IP
 Address. The RDT uses configuration information for the Destination
 IP Address.

The present invention reduces unnecessary overhead
 20 information on bandwidth limited RF channels and improves data
 transmission efficiency/speed over the RF Data Network. A typical,
 effective data rate for the Radio Network Data Link Layer is
 approximately 3400 bits per second (bps) across the allowable
 datagram sizes and expected RF signaling environments. At 3400
 25 bps, the 10 bytes of IP Header information per datagram eliminated
 on the RF Network results in an average savings of 24msec per

datagram. In some situations, this 10 bytes is the difference between transmitting two data calls rather than one data call in the same time period. Time savings per call in such instances may be for example on the order of 700ms.

5

Thus, the present invention permits a proprietary radio network layer protocol that is uniquely desirable in an RF environment to be used with and support standard IP features. As a result, standard computer communications protocols can be used with
10 a radio data network while minimizing bandwidth use on the radio channel and using a protocol optimized (hardened) for the RF environment.

The following is a description of a transmitting message from a
15 host computer in the wireline Ethernet network to a radio data terminal (RDT). First, the host computer looks up the RDT's IP address in its routing table and finds the IP address of the data gateway's central activity processor listed as the next gateway for the RF data network. The host computer then forwards the message to
20 the central activity processor (CAP) using its Ethernet address. If the host does not know the central activity processor's Ethernet address, it uses address resolution protocol (ARP) to ask the CAP for its address. Second, the central activity processor converts the IP Header to the Radio Network Layer Header and forwards the
25 message to a radio system interface which converts the destination IP address either to a radio logical ID (LID) or group ID (GID).

A flowchart which describes the conversion of the standard IP header into the modified radio network layer header (performed at data gateway 40) will now be described in conjunction with the flowchart shown in Figure 7. The IP datagram is fragmented if the
5 single IP datagram will not fit in one frame of the physical radio network, e.g., 512 bytes (block 100). The more fragments bit (MF) and fragment offset field are set in the RNL header based on results of that fragmentation (block 102). The VERS field bits are set to the
10 UNUSED field bits are set to zero (block 104). The IDENTIFICATION and PROTOCOL fields from the IP header are copied into the RNL header (block 106). The SOURCE IP address from the IP header is then copied to the EXTENDED ADDRESS of the RNL header (block 108). A decision is made in block 110 if there
15 are more fragments in the datagram. If there are, control proceeds back to process the next fragment in block 102; otherwise, the procedure is completed.

Thus, only a minimum number of fields from the standard IP
20 header are used to formulate the RNL header. A large number of the standard IP header fields are simply omitted.

The radio system interface then sends the message to a radio or group of radios via the data interface module 38, site 22 or 23, and
25 an assigned RF working channel. The radio/radio data interface sends the message to the radio data terminal using a transfer command.

While the radio data terminal may be configured to operate on the received message without any conversion of the radio network layer header, in a preferred embodiment of the present invention, the network driver of the radio data terminal (see Figure 6), converts the RNL header back into the standard IP header format. This conversion is performed at the RDT to make the RDT compatible with the wide variety of software products written based on standard TCP/UDP (transport layer) and IP (network layer) protocols. These software products are normally compliant with the popular WINSOCKET™ interface standard from Microsoft. WINSOCKET™ defines an interface between the applications layer and the TCP/UDP layer. Application software products compliant with WINSOCKET™ need not be aware or concerned that the RDT is a part of a radio network. Except for reduced throughput, the operator can treat the RDT like any other conventional PC connected to a wireline network.

Figure 8 is a flowchart diagram illustrating the conversion of the modified RNL header to the standard IP header at the RDT. The UNUSED bit field is set to zero (block 112), and the VERSION, HEADER LENGTH, SERVICE TYPE, DF bit, and TIME TO LIVE FIELDS are set to their static values as described above which are stored in the RDT. Next, the DESTINATION IP address is set to the IP address configured for this particular RDT (block 116). The IDENTIFICATION, MF bit, FRAGMENT OFFSET, and PROTOCOL fields from the RNL header are copied into the reconstructed IP header (block 118). The EXTENDED ADDRESS from the RNL is copied into the SOURCE ADDRESS field of the reconstructed IP header (block 120). The TOTAL LENGTH field of the IP header is

set based upon the length already present in the data link layer plus ten bytes (block 122). The HEADER CHECKSUM is calculated and inserted into the reconstructed IP header (block 124).

- 5 A message from a radio data terminal to a host computer on the Ethernet is processed as follows. First, the radio data terminal sends the message to the radio/radio data interface using a transfer command. The radio network layer header contains the IP address of the host computer. The ^{source}~~destination~~ radio address corresponds to one
10 of the IDs in a block of IDs stored at the data gateway 40.

M.A. 7/28/97
P.H. 7/28/97
EJ 8/6/97

- The flowchart in Figure 9 illustrates the conversion of the standard IP header into the radio network layer header at the RDT for a message going from the RDT to the Ethernet. The VERS field
15 bits are set to the version of ^{the}~~ten~~ RNL in use by the RDT, and the UNUSED field bits are set to zero (130). The IDENTIFICATION, MF bit, FRAGMENT OFFSET, and PROTOCOL fields are copied
20 into the EXTENDED ADDRESS of the RNL header (block 134).

M.A. 7/27/97
D.A. 7/28/97
P.H. 7/28/97
EJ 8/6/97

- A radio system interface receives the message in from the radio transmitted over an assigned RF working channel and routes that message to the central activity processor using the IP address in the
25 radio network layer header. The conversion of the radio network layer header to the IP header at the data gateway is illustrated in the flowchart shown in Figure 10.

The UNUSED bit is set to zero (block 140), and the VERSION, HEADER LENGTH, SERVICE TYPE, DF bit, and TIME TO LIVE fields are set to their static values as described above and stored in the data gateway (block 142). The SOURCE IP ADDRESS is set to
5 the EXTENDED IP ADDRESS configured for this particular RDT (block 144). The IDENTIFICATION, MF bit, FRAGMENT OFFSET, and PROTOCOL fields from the RNL header are copied into the reconstructed IP header (block 146). The extended address from the RNL header is copied into the DESTINATION ADDRESS of the
10 standard IP header (block 148). The TOTAL LENGTH is determined from the LENGTH field in the data link layer plus ten bytes (block 150). The HEADER CHECKSUM is calculated and inserted into the reconstructed standard IP header (block 152). The central activity processor then forwards the message using the host's Ethernet
15 address. Again, if the central activity processor does not know the host's Ethernet address, it uses address resolution protocol to ask the host for such address.

While the invention has been described in connection with
20 what is presently considered to be the most practical and preferred embodiment, it is to be understood that the invention is not to be limited to the disclosed embodiment, but on the contrary, is intended to cover various modifications and equivalent arrangements included within the spirit and scope of the appended claims.

WHAT IS CLAIMED IS:

1. In a packet-based communications system, each packet including a header and an associated data portion, where packeted messages are communicated between devices on first and second networks, the first network using a standard internetwork protocol (IP), a method comprising the steps of:

- 5 (a) transmitting a message from a first device on the first network to a second device on the second network, and
- (b) omitting one or more predetermined fields from the standard IP header of each packet in the message to obtain a modified header before transmitting the message over the second network to the second device.

10

2. The method in claim 1, further comprising the steps of:

- (c) transmitting another message from the second device to the first device, and
- (d) adding one or more predetermined fields to the modified header of each
- 15 packet in the other message to convert that modified header into a corresponding standard IP header before transmitting the other message over the first network.

3. The method in claim 2, wherein the adding step (d) includes reconstructing the corresponding standard IP header by adding known static
- 20 standard IP protocol header fields to fields from the modified header of each packet in the other message along with dynamic fields of the standard IP header determined from information in that modified header.

4. The method in claim 3, wherein the determined dynamic fields include source and destination address fields and a header checksum field.

5. The method in claim 1, wherein the second network is a radio
5 communications network having RF communications channels.

6. The method in claim 1, wherein fields omitted in the modified header include one or more of the following fields:
header length, service type, total length, time to live, header checksum,
10 option, and padding.

7. The method in claim 1, wherein address fields in the standard IP header are converted into a single extended address field included in the modified header corresponding to the source address from the standard IP header.
15

8. The method in claim 2, wherein fields added to the modified header include one or more of the following static fields:
header length, service type, time to live, option, and padding.

20 9. The method in claim 2, wherein fields added to the modified header include one or more of the following determined, dynamic fields:
total length, source address, destination address, and header checksum.

10. The method in claim 1, further comprising:
converting the modified header of each packet in the message upon receipt
of the message at the second device into the standard IP header.

5 11. The method in claim 10, further comprising:
omitting at the second device one or more fields from the standard IP header
of each packet of a message to obtain the modified header before transmitting the
message intended for the first device over the second network.

10 12. A communications system permitting first and second data processing
units associated respectively with a first network and a second network, the second
network having more limited communications bandwidth than the first network, to
transmit packets of information, each packet including a header portion and a data
portion, wherein certain fields in the header portion of a packet from the first
15 network are omitted before transmitting the packet over the second network.

13. The system in claim 12, wherein the second network is an radio
communications network having RF communications channels.

20 14. The system in claim 13, wherein a modified radio network header is
constructed based on information in an RF communications protocol and
information in the header portion.

25 15. The system in claim 12, wherein certain predetermined fields of the
header portion of a packet from the first network are stored and used to reconstruct a

header portion of a packet transmitted from the second network to the first network into a header format used on the first network.

16. The system in claim 15, wherein the first network uses a standard
5 internetwork protocol (IP) header and the second network uses a radio network header adapted for use in a radio communications network.

17. A method for generating a modified network layer header adapted from
a standard network layer header by omitting certain fields included in the standard
10 network layer header.

18. The method in claim 17, wherein the modified network layer header
includes one or more of the following fields: a version field, an identification field,
a more fragments field, a fragment offset field, an address field, and a protocol field.
15

19. A packet-based communications system, each packet including a header
and an associated data portion, comprising:

first and second communications networks each having at least one device
for transmitting and receiving messages, the first network using a standard
20 internetwork communications protocol and the second network using a different network communications protocol, and
a data gateway connecting the first and second networks,
wherein for messages received from the first network intended for the
second network, the data gateway deletes certain fields of a standard header for each
25 message packet corresponding to the standard internetwork communications protocol to obtain a modified header, and for messages received from the second

network intended for the first network, the data gateway adds fields to the modified header to obtain a standard header.

20. The system in claim 19, wherein the first network is a wireline network
5 including a plurality of computers connected to a wireline communications bus and the second network is an RF network including a plurality of radio units that communicate over RF communications channels using a central controller.

21. The system in claim 20, wherein the data gateway includes a central
10 processor connected to the wireline network and to a radio system interface connected to the central controller.

22. The system in claim 21, wherein each radio unit includes a radio data
terminal connected to a radio data interface and radio transceiving circuitry, the
15 radio transceiving circuitry transceiving control information over an RF control channel and transceiving message data over an RF working channel assigned by the central controller.

23. The system in claim 19, wherein fields omitted in the modified header
20 include one or more of the following fields:

header length, service type, total length, time to live, header checksum,
option, and padding.

24. The system in claim 19, wherein source and destination address fields in
25 the standard header are converted into a single extended address field included in the modified header corresponding to the source address from the standard header.

25. The system in claim 23, wherein fields added to the modified header include one or more of the following static fields:

header length, service type, time to live, option, and padding.

5 26. The system in claim 23, wherein fields added to the modified header include one or more of the following determined fields:

total length, source address, and header checksum.

10 27. The system in claim 22, wherein the central processor converts a destination IP address to a radio address for packets transmitted to the radio network and generates the destination IP address using an address of a radio sending a message to the wireline network.

15 28. The system in claim 22, wherein for received messages, network layer software in the radio data terminal converts modified headers into the standard headers, and for messages to be transmitted, the network layer software converts standard headers into modified headers.

20 29. A method for converting a standard internetwork protocol (IP) header into a modified header occupying a smaller bandwidth, comprising the steps of:

(a) copying identification and protocol fields from the IP header into the modified header, and

(b) copying a source IP address from the IP header into the address field of the modified header,

25 wherein other fields in the IP header are not included in the modified header.

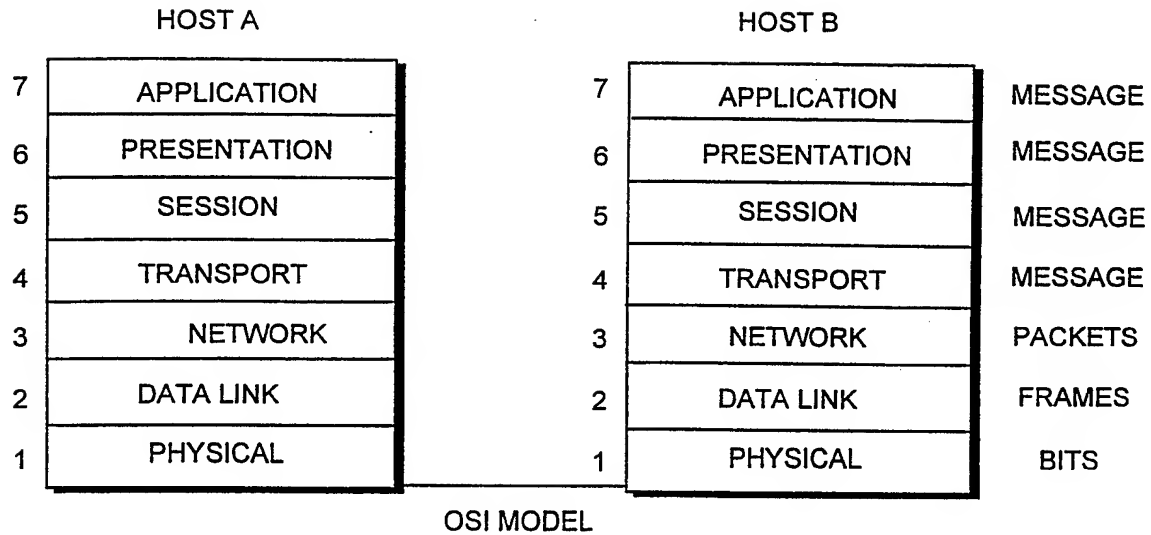
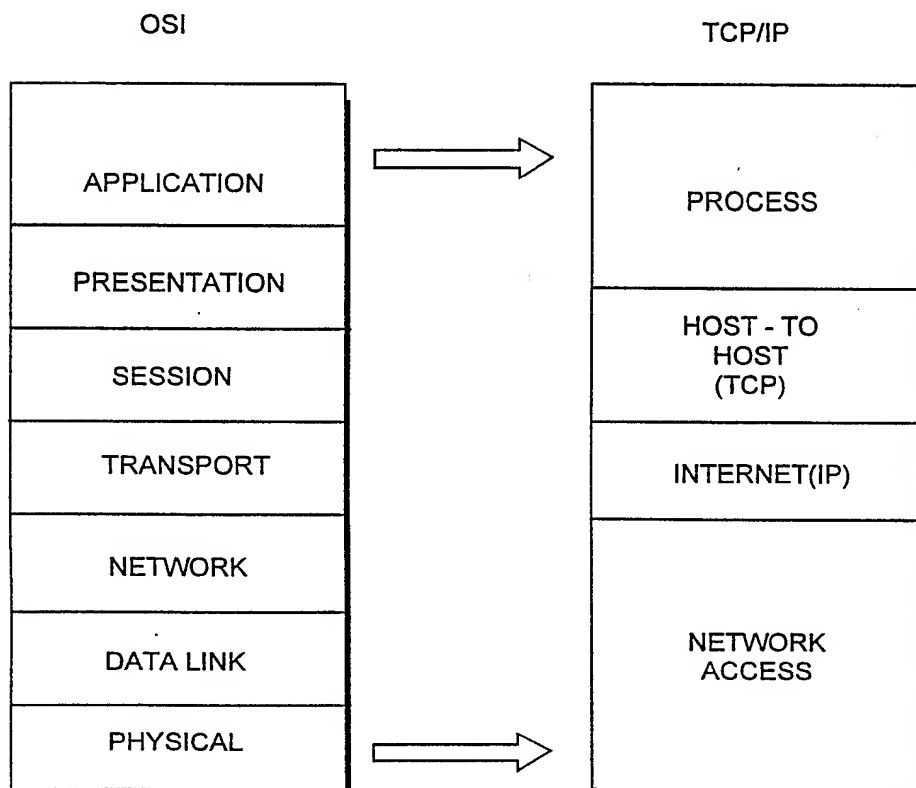
30. The method in claim 29, further comprising before step (a) the steps of:
fragmenting an IP datagram if the IP datagram exceeds a maximum frame
length, and

5 inserting values for a more fragments field and a fragment offset field into
the modified header based on the fragmenting step.

31. The method in claim 29, further comprising the steps of:
storing static field values from the IP header, and
reconstructing the IP header from the modified header and the stored static
10 field values.

32. The method in claim 31, wherein the static fields include:
header length, service type, and time to live fields.

15 33. The method in claim 31, the reconstructing step further comprising:
copying identification, more fragments, fragment offset, and protocol fields
from the modified header into the reconstructed IP header,
copying an address from the modified header to the source address field of
the IP header;
20 determining a total length, header checksum, and destination address fields
of the IP header from information provided on the modified header.

**FIG. 1A****FIG. 1B**

2/9

STANDARD INTERNET PROTOCOL (IP) DATAGRAM**FIG. 2A****STANDARD INTERNET PROTOCOL (IP) HEADER**

0	4	8	16	19	24	31
VERS	HLEN	SERVICE TYPE	TOTAL LENGTH			
IDENTIFICATION			FLAGS	FRAGMENT OFFSET		
TIME TO LIVE		PROTOCOL	HEADER CHECKSUM			
SOURCE IP ADDRESS						
DESTINATION IP ADDRESS						
IP OPTIONS (IF ANY)					PADDING	
DATA						
...						

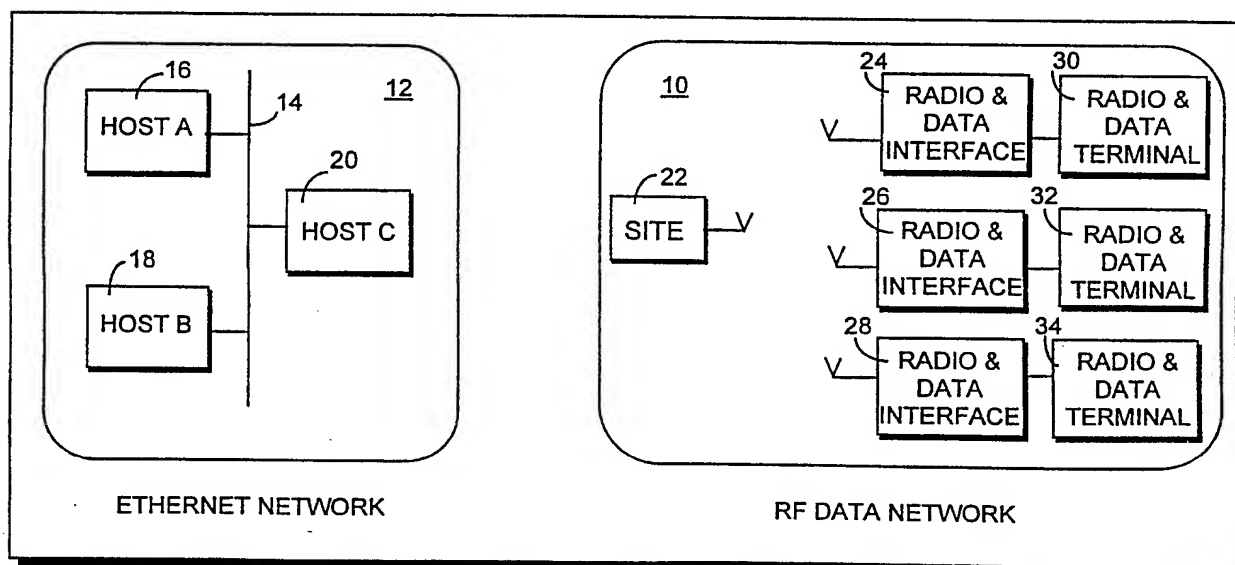
FIG. 2B

FIG. 3
SUBSTITUTE SHEET (RULE 26)

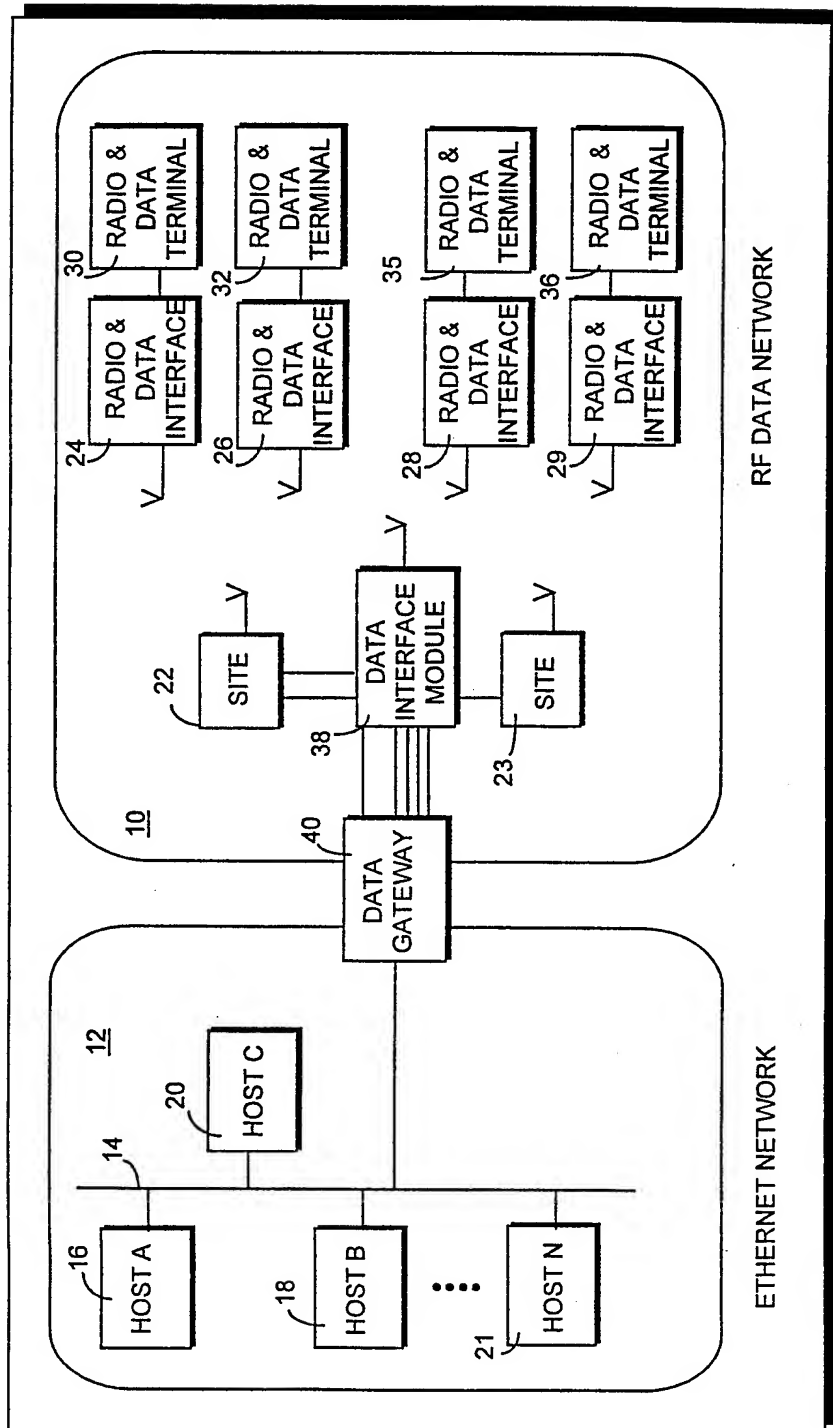
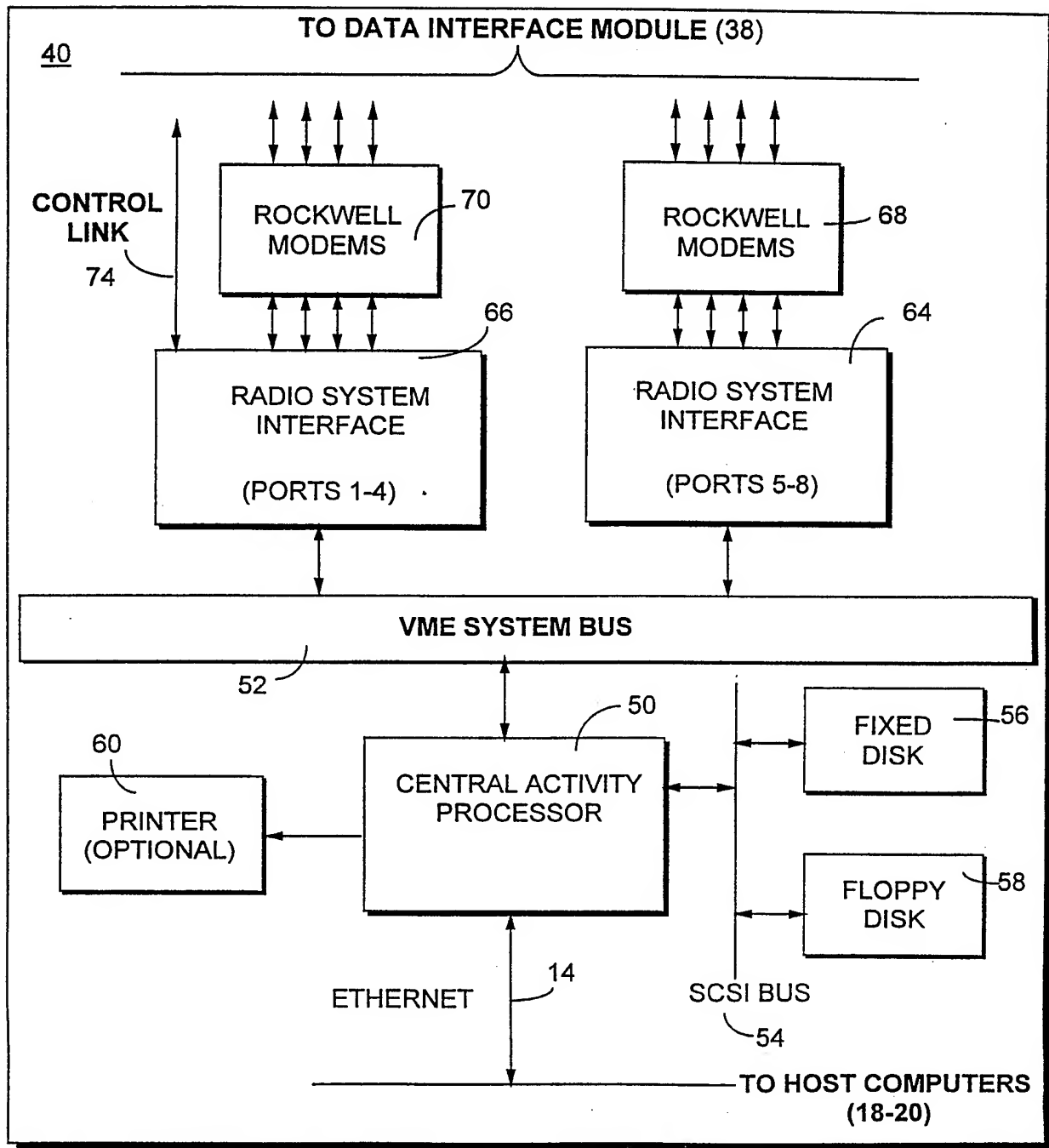


FIG. 4

**FIG. 5**

SUBSTITUTE SHEET (RULE 26)

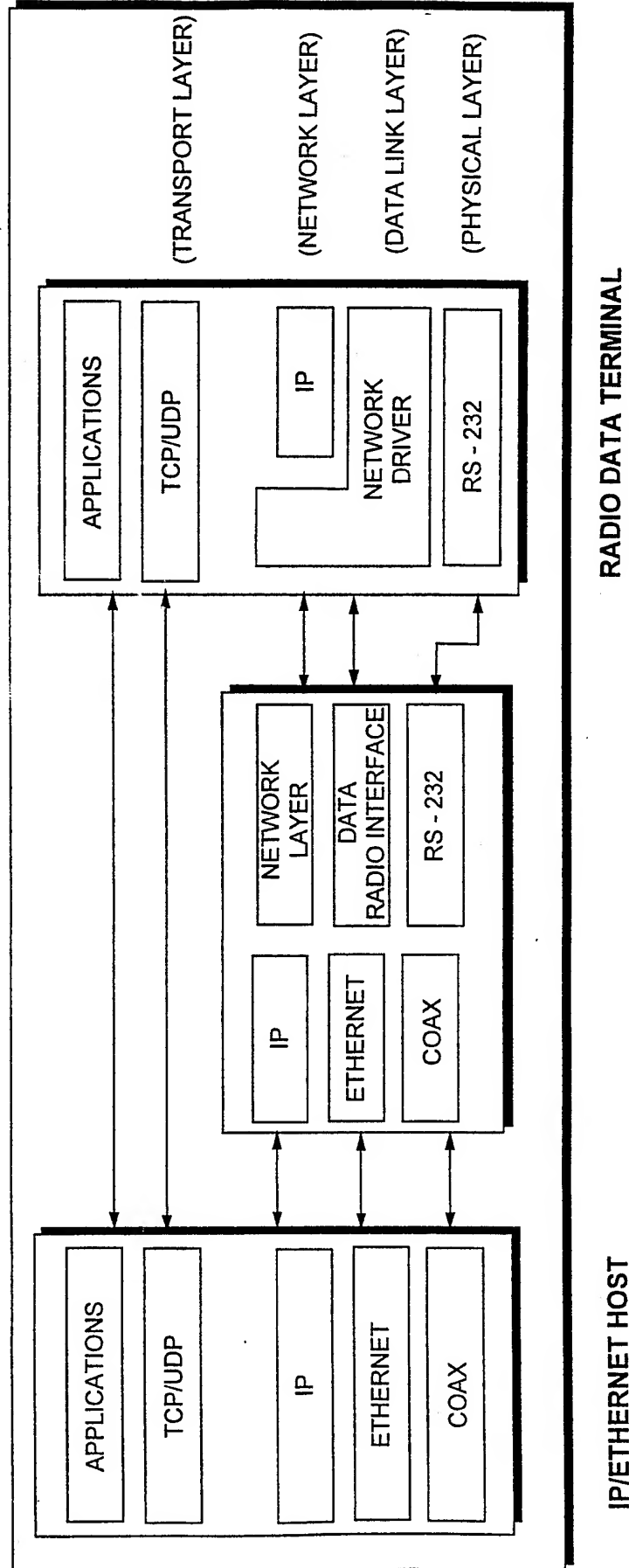


FIG. 6

6/9

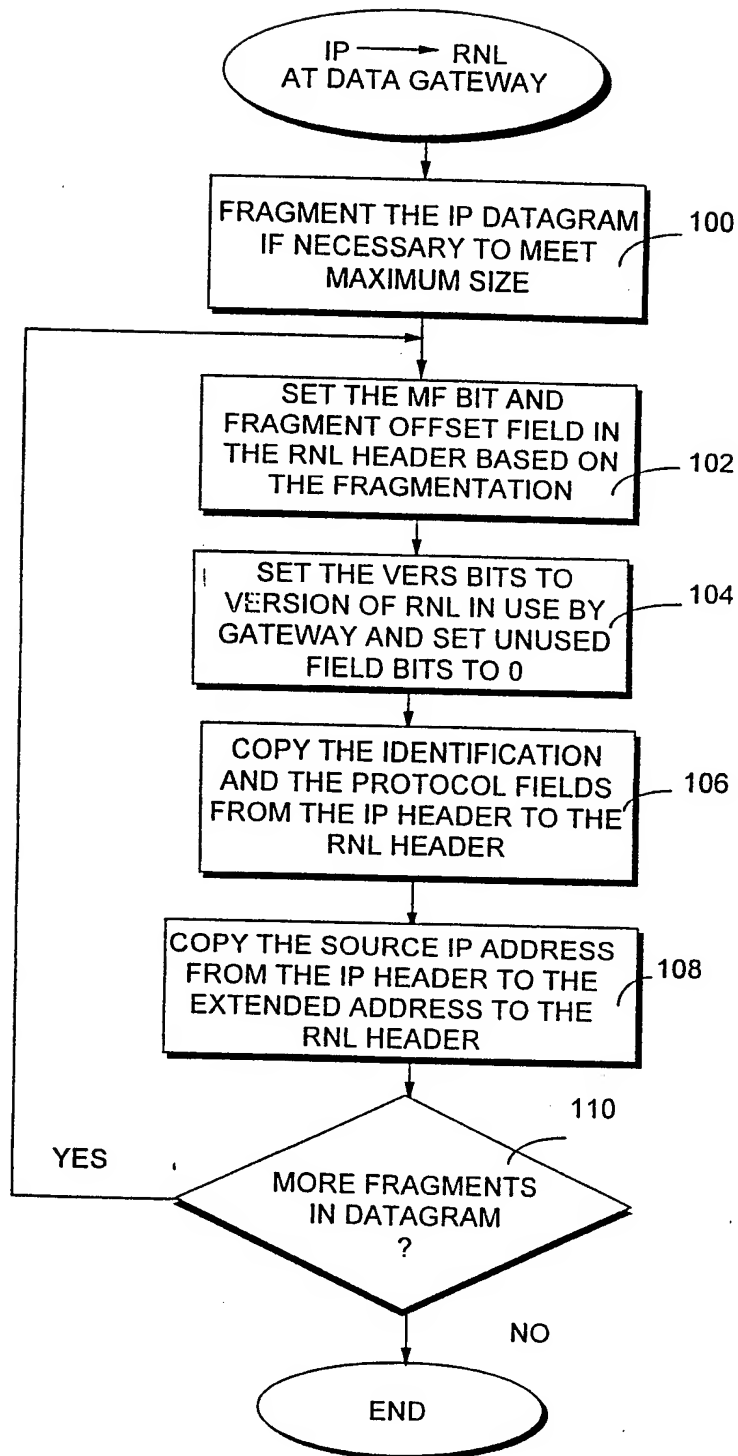


FIG. 7

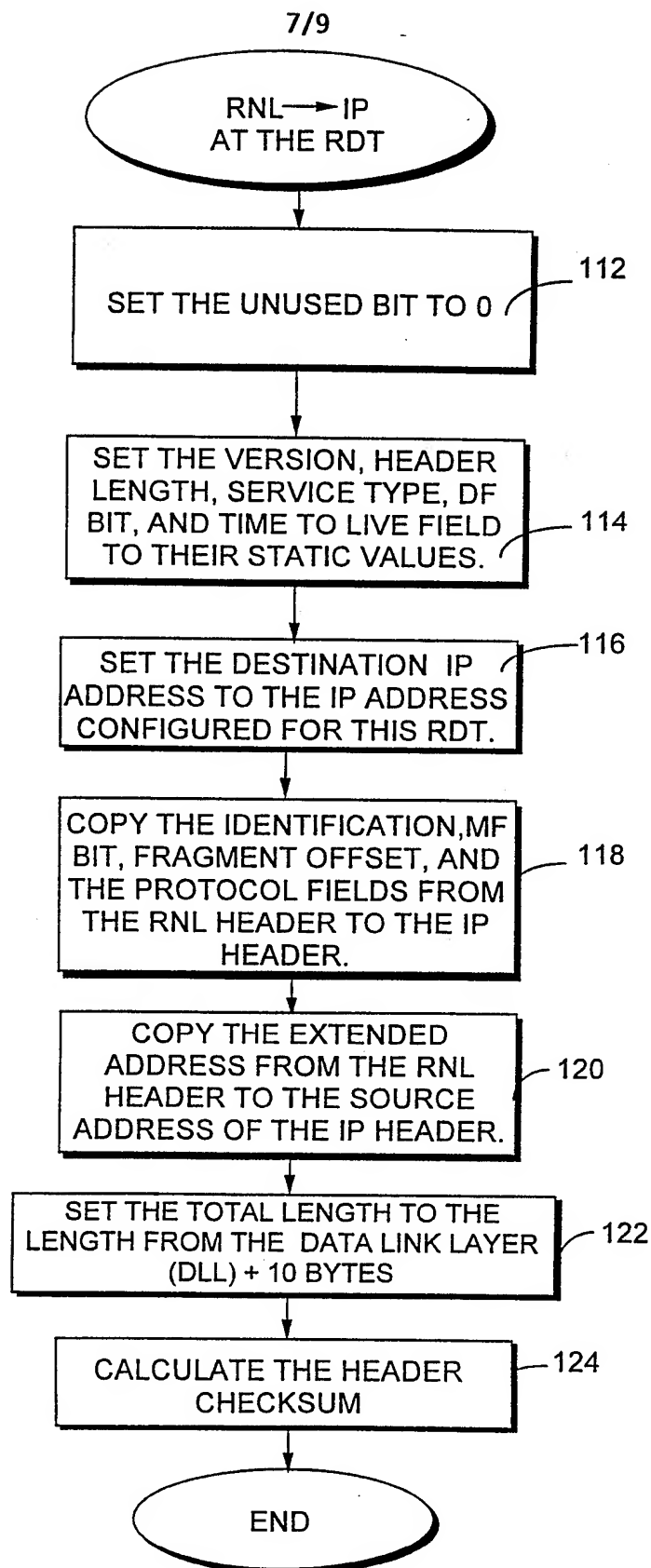
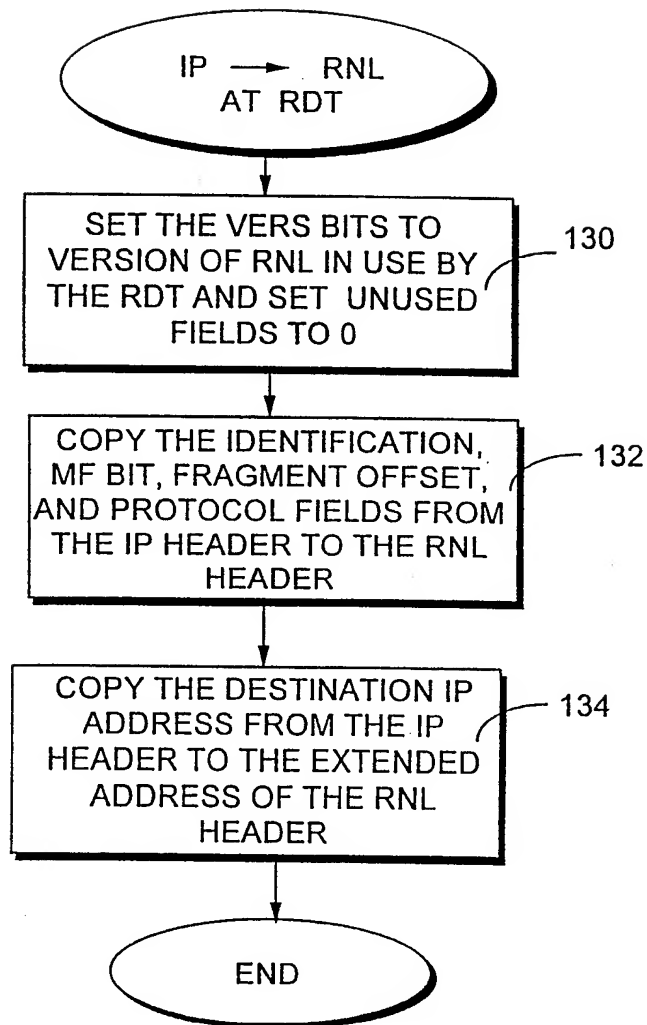


FIG. 8
SUBSTITUTE SHEET (RULE 26)

8/9

**FIG. 9**

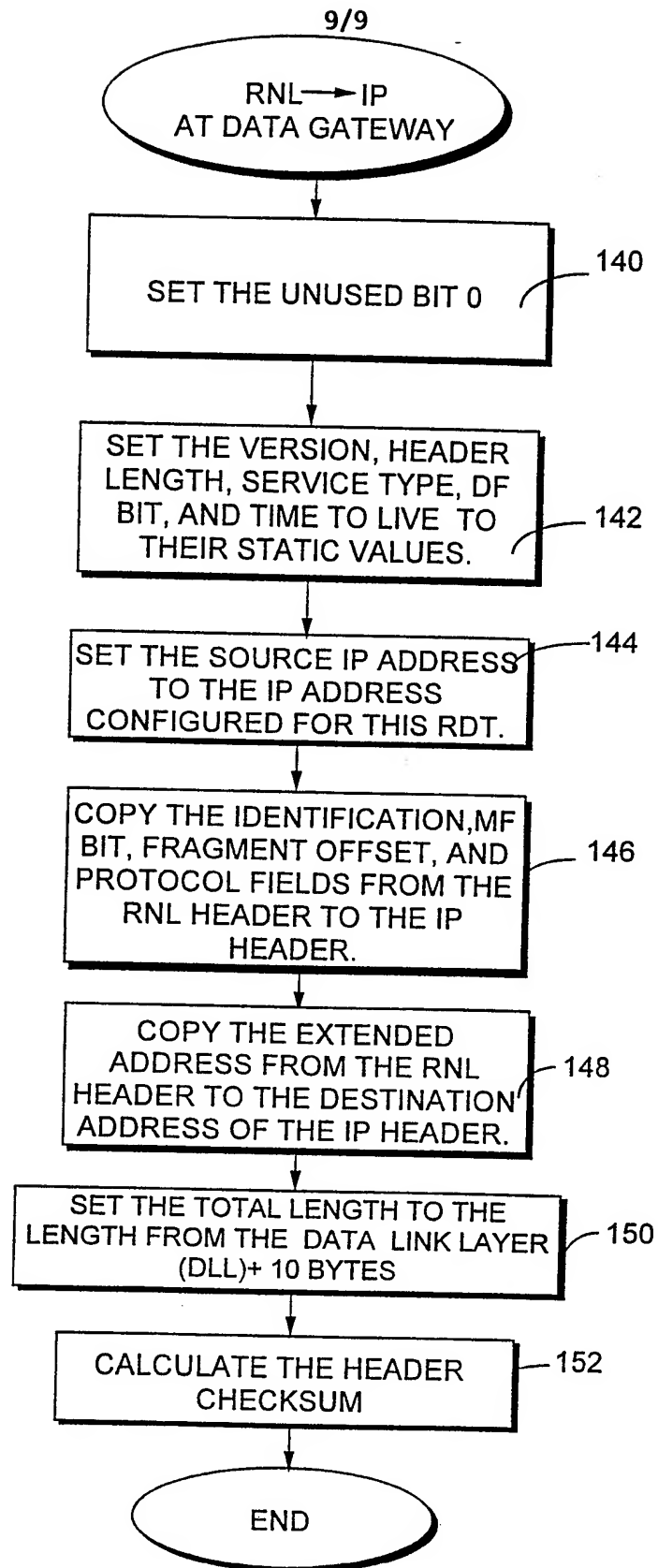


FIG. 10
SUBSTITUTE SHEET (RULE 26)